AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Original) An authenticated-encryption method that uses a key, a nonce and an n-bit block cipher to encrypt a message into a ciphertext, the method comprising:

partitioning the message into a message body comprising a sequence of n-bit message blocks, and a message fragment of at most n bits;

generating a sequence of offsets from the nonce and the key;

computing a ciphertext body using the block cipher, the message body, the key, the nonce, and the sequence of offsets;

computing a ciphertext fragment using the block cipher, the message fragment, the key, and an offset;

computing a tag as a function of the message body, the message fragment, the sequence of offsets, and the key; and

defining the ciphertext to include the ciphertext body, the ciphertext fragment, and the tag.

2. (Original) The method of claim 1, wherein generating the sequence of offsets involves:

determining a first offset as a function of the nonce and the key; and

determining each subsequent offset by combining a previous offset and a basis offset, wherein each basis offset is determined as a function of the key.

3. (Original) The method of claim 1, wherein generating the sequence of offsets involves determining an offset by combining a base offset and a fixed offset, wherein the base offset is a function to the key and the nonce, and the fixed offset is a function of the key and the position of the offset in a sequence of offsets.

4. (Original) The method of claim 1, wherein generating the sequence of offsets involves:

generating a sequence of fixed offsets from the key;

generating a base offset from the key and the nonce;

generating a sequence of translated offsets by combining each fixed offset with the base offset to get a corresponding translated offset; and

using the sequence of translated offsets as the sequence of offsets.

5. (Original) The method of claim 4, wherein the key determines a sequence of basis offsets and each fixed offset is determined by xoring some combination of basis offsets.

6. (Original) The method of claim 5, wherein each basis offset except for the first basis offset is determined by a shift and a conditional xor applied to a previous basis offset.

7. (Original) The method of claim 5, wherein the order that basis offsets are combined into fixed offsets is determined according to a Gray code.

8. (Original) The method of claim 1, wherein generating the sequence of offsets involves:

computing a sequence of basis offsets from the key;

computing a base offset from the key and the nonce; and

computing the first offset in the sequence of offsets as a function of the base offset, the key, and the nonce, and computing each subsequent offset in the sequence of offsets by combining the prior offset with a basis offset.

9. (Original) The method of claim 1, wherein generating the sequence of offsets involves:

computing a key-variant by enciphering a constant with the block cipher, wherein the block cipher is keyed by the given key; and

computing the sequence of offsets as a function of the key variant and the nonce.

10. (Original) The method of claim 1, wherein computing the ciphertext body involves:

combining each message block in the message body with a corresponding offset to produce a corresponding input block;

applying the block cipher to each input block to produce a corresponding output block;

combining each output block with a corresponding offset to produce a corresponding ciphertext block; and

concatenating the ciphertext blocks to determine the ciphertext body.

11. (Original) The method of claim 1, wherein computing the ciphertext fragment involves:

computing a precursor pad as a function of an offset and the length of the message;

computing a pad by applying the block cipher to the precursor pad; and

computing the ciphertext fragment by combining the message fragment and the pad.

12. (Original) The method of claim 1, wherein computing the tag involves:

computing a checksum as a function of the message, the ciphertext fragment, and the sequence of offsets; and

computing the tag as a function of the checksum, the key, and an offset.

13. (Original) The method of claim 1, wherein computing the tag involves:

computing a checksum from at least the message ;

combining the checksum with an offset to produce a precursor full tag;

computing a full tag by applying the block cipher to the precursor full tag; and

computing a tag as a portion of the full tag.

14. (Original) An authenticated-encryption method that uses a key, a nonce, and an n-bit block cipher to decrypt a ciphertext into a message or a message-invalid signal, the method comprising:

partitioning the ciphertext into a ciphertext body comprising a sequence of n-bit ciphertext blocks, a ciphertext fragment of at most n bits, and a tag;

generating a sequence of offsets from the nonce and the key;

computing a message body using the block cipher, the ciphertext body, the key, the nonce, and the sequence of offsets;

computing a message fragment using the block cipher, the ciphertext fragment, the key, and an offset;

computing a new tag as a function of the message body, the message fragment, the sequence of offsets, the block cipher, and the key; and

comparing the new tag with the tag;

if the new tag matches the tag, returning the message, wherein the message includes the message body and the message fragment; and

if the new tag does not match the tag, returning a message-invalid signal.

15. (Original) The method of claim 14, wherein generating the sequence of offsets involves:

generating a sequence of fixed offsets from the key;

generating a base offset from the key and the nonce;

generating a sequence of translated offsets by combining each fixed offset with the base offset to get a corresponding translated offset; and

using the sequence of translated offsets as the sequence of offsets.

16. (Original) The method of claim 14, wherein computing the message body involves:

combining each ciphertext block in the ciphertext body with a corresponding offset to produce a corresponding output block;

applying the block-cipher inverse to each output block to produce a corresponding input block;

combining each input block with a corresponding offset to produce a corresponding message block; and

defining the message body to be the sequence of message blocks.

17. (Original) The method of claim 14, wherein computing the message fragment involves:

computing a precursor pad as a function of an offset and the length of the ciphertext;

computing a pad by applying the block cipher to the precursor pad; and

computing the message fragment by combining the ciphertext fragment and the pad.

18. (Original) The method of claim 14, wherein computing the tag involves:

computing a checksum as a function of at least the message; and

computing the tag as a function of the checksum, the key, and an offset.

19. (Original) A computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform an authenticated-encryption method that uses a key and a nonce to encrypt a message into a ciphertext, the method comprising:

partitioning the message into a message body including a sequence of n-bit message blocks, and a message fragment of at most n bits;

generating a sequence of offsets from the nonce and the key;

computing a ciphertext body using a block cipher, the message body, the key, the nonce, and the sequence of offsets;

computing a ciphertext fragment using the block cipher, the message fragment, the key, and an offset;

computing a tag as a function of the message body, the message fragment, the sequence of offsets, and the key; and

defining the ciphertext to include the ciphertext body, the ciphertext fragment, and the tag.

20. (Original) The computer-readable storage medium of claim 19, wherein generating the sequence of offsets involves:

determining a first offset as a function of the nonce and the key; and

determining each subsequent offset by combining a previous offset and a basis offset, wherein each basis offset is determined as a function of the key.

21. (Original) The computer-readable storage medium of claim 19, wherein generating the sequence of offsets involves determining an offset by combining a base offset and a fixed offset, wherein the base offset is a function to

the key and the nonce, and the fixed offset is a function of the key and a position of the fixed offset in a sequence of fixed offsets.

22. (Original) The computer-readable storage medium of claim 19, wherein generating the sequence of offsets involves:

generating a sequence of fixed offsets from the key;

generating a base offset from the key and the nonce;

generating a sequence of translated offsets by combining each fixed offset with the base offset to get a corresponding translated offset; and

using the sequence of translated offsets as the sequence of offsets.

23. (Original) The computer-readable storage medium of claim 22, wherein the key determines a sequence of basis offsets and each fixed offset is determined by xoring some combination of basis offsets.

24. (Original) The computer-readable storage medium of claim 23, wherein each basis offset except for the first basis offset is determined by a shift and a conditional xor applied to a previous basis offset.

25. (Original) The computer-readable storage medium of claim 24, wherein the order that basis offsets are combined into fixed offsets is determined according to a Gray code.

26. (Original) The computer-readable storage medium of claim 19, wherein generating the sequence of offsets involves:

computing a sequence of basis offsets from the key;

computing a base offset from the key and the nonce; and

computing a sequence of translated offsets, wherein the first offset is determined from the base offset, the key, and the nonce, and subsequent offsets are determined by combining the prior translated offset with a basis offset.

27. (Original) The computer-readable storage medium of claim 19, wherein generating the sequence of offsets involves:

computing a key-variant offset by enciphering a constant with the block cipher, wherein the block cipher is keyed by a given key; and

computing the sequence of offsets using the key-variant offset.

28. (Original) The computer-readable storage medium of claim 19, wherein computing the ciphertext body involves:

combining each message block in the message body with a corresponding offset to produce a corresponding input block;

applying the block cipher to each input block to produce a corresponding output block; and

combining each output block with a corresponding offset to produce a corresponding ciphertext block.

29. (Original) The computer-readable storage medium of claim 19, wherein computing the ciphertext fragment involves:

computing a precursor pad as a function of an offset;

computing a pad by applying the block cipher to the precursor pad; and

computing the ciphertext fragment by combining the message fragment and the pad.

30. (Original) The computer-readable storage medium of claim 19, wherein computing the tag involves:

computing a checksum as a function of the message and a sequence of offsets; and

computing the tag as a function of the checksum, the key, and an offset.

31. (Original) The computer-readable storage medium of claim 19, wherein computing the tag involves:

computing a checksum from the message blocks, the message fragment, and a pad;

combining the checksum with an offset to produce a precursor full tag;

computing a full tag by applying the block cipher to the precursor full tag; and

computing a tag as a portion of the full tag.

32. (Original) A computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform an authenticated-encryption method that uses a key and a nonce to decrypt a ciphertext into a message, the method comprising:

partitioning the ciphertext into a ciphertext body including a sequence of n-bit ciphertext blocks, a ciphertext fragment of at most n bits, and a tag;

generating a sequence of offsets from the nonce and the key;

computing a message body using a block cipher, the ciphertext body, the key, the nonce, and the sequence of offsets;

computing a message fragment using the block cipher, the ciphertext fragment, the key, and an offset;

computing a new tag as a function of the message body; and

comparing the new tag with the tag;

if the new tag matches the tag, returning the message, wherein the message includes the message body and the message fragment; and

otherwise, if the new tag does not match the tag, returning a message invalid signal.

33. (Original) The computer-readable storage medium of claim 32, wherein generating the sequence of offsets involves:

generating a sequence of fixed offsets from the key;

generating a base offset from the key and the nonce;

generating a sequence of translated offsets by combining each fixed offset with the base offset to get a corresponding translated offset; and

using the sequence of translated offsets as the sequence of offsets.

34. (Original) The computer-readable storage medium of claim 32, wherein computing the message body involves:

combining each ciphertext block in the ciphertext body with a corresponding offset to produce a corresponding input block;

applying the block cipher to each input block to produce a corresponding output block; and

combining each output block with a corresponding offset to produce a corresponding message block.

35. (Original) The computer-readable storage medium of claim 32, wherein computing the message fragment involves:

computing a precursor pad as a function of an offset;

computing a pad by applying the block cipher to the precursor pad; and

computing the message fragment by combining the ciphertext fragment and the pad.

36. (Original) The computer-readable storage medium of claim 32, wherein computing the tag involves:

computing a checksum as a function of the message body; and

computing the tag as a function of the checksum, the key, and an offset.

37. (Original) An authenticated-encryption apparatus that uses a key and a nonce to encrypt a message into a ciphertext, the apparatus comprising:

a partitioning mechanism that is configured to partition the message into a message body including a sequence of n-bit message blocks, and a message fragment of at most n bits;

an offset generating mechanism that is configured to generate a sequence of offsets from the nonce and the key;

an enciphering mechanism that is configured to compute a ciphertext body using a block cipher, the message body, the key, the nonce, and the sequence of offsets;

wherein the enciphering mechanism is additionally configured to compute a ciphertext fragment using the block cipher, the message fragment, the key, and an offset;

a tag computing mechanism that is configured to compute a tag as a function of the message body, the message fragment, the sequence of offsets, and the key; and

an assembly mechanism that is configured to define the ciphertext to include the ciphertext body, the ciphertext fragment, and the tag.

38. (Original) An authenticated-encryption apparatus that uses a key and a nonce to decrypt a ciphertext into a message, the apparatus comprising:

a partitioning mechanism that is configured to partition the ciphertext into a ciphertext body including a sequence of n-bit ciphertext blocks, a ciphertext fragment of at most n bits, and a tag;

an offset generating mechanism that is configured to generate a sequence of offsets from the nonce and the key;

a deciphering mechanism that is configured to compute a message body using a block cipher, the ciphertext body, the key, the nonce, and the sequence of offsets;

wherein the deciphering mechanism is configured to compute a message fragment using the block cipher, the ciphertext fragment, the key, and an offset;

a tag computing mechanism that is configured to compute a new tag as a function of the message body; and

a comparison mechanism that is configured to compare the new tag with the tag;

wherein if the new tag matches the tag, the apparatus is configured to return the message, wherein the message includes the message body and the message fragment; and

wherein if the new tag does not match the tag, the apparatus is configured to return a message invalid signal.

39. (Original) An authenticated-encryption method that uses an n-bit block cipher, a key, and an n-bit nonce to encrypt a message into a ciphertext, the method comprising:

partitioning the message into m message blocks and one final fragment, each message block having n bits and the final fragment having between 0 and n bits;

using the block cipher, the key, and the nonce to generate a sequence of m offsets, each offset having n bits;

using the block cipher, the key, the nonce, and the length of the message to generate an n-bit final offset;

for each number i between 1 and m, xoring the i^{th} message block with the i^{th} offset to determine an i^{th} input block;

for each number i between 1 and m, applying the block cipher, keyed by the key, to the i^{th} input block, to determine an i^{th} output block;

for each number i between 1 and m, xoring the i^{th} output block with the i^{th} offset to determine an i^{th} ciphertext block;

concatenating the m ciphertext blocks to determine a ciphertext body;

computing an encoded length by encoding the length of the final fragment as an n-bit string;

xoring the encoded length with the final offset to determine a precursor pad;

computing a pad by applying the block cipher, keyed by the key, to the precursor pad;

xoring the final fragment with a portion of the pad to determine a ciphertext fragment having the same length as the final fragment;

computing a padded ciphertext fragment by appending to the ciphertext fragment a sufficient number of zero bits so that the padded ciphertext fragment has n bits;

computing a checksum by xoring together the m message blocks, the pad, and the padded ciphertext fragment;

computing a precursor full tag by xoring together the checksum and the m^{th} offset;

determining a full tag by applying the block cipher, keyed by the key, to the precursor full tag;

computing a tag as a portion of the full tag; and

defining the ciphertext to be the ciphertext body, the ciphertext fragment, and the tag.

40. (Original) The method of claim 39, wherein the i^{th} offset from the sequence of offsets is determined by:

computing a 0^{th} basis offset by applying the block cipher, keyed by the key, to a constant;

for each positive number i, defining the i^{th} basis offset from the prior basis offset by shifting the prior basis offset left one position, and then xoring the resulting value with a constant that depends on the first bit of the prior basis offset;

computing a base offset by applying the block cipher, keyed by the key, to the xor of the 0^{th} basis offset and the nonce;

defining the first offset in the sequence of offsets as the xor of the 0^{th} basis offset and the base offset; and

for each integer i greater than one, defining the i^{th} offset in the sequence of offsets as the xor of the prior offset and the j^{th} basis offset, where j is the number of zero-bits following the last one-bit when the number i is written in binary.

41. (Original) The method of claim 39, wherein the final offset is determined by shifting the 0^{th} basis offset one position to the right, xoring a constant that depends on the last bit of the 0^{th} basis offset, and then xoring the m^{th} offset.

42-49 (Canceled).

50. (Original) A computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform an

authenticated-encryption method that uses an n-bit block cipher, a key, and an n-bit nonce to encrypt a message into a ciphertext, the method comprising:

partitioning the message into m message blocks and one final fragment, each message block having n bits and the final fragment having between 0 and n bits;

using the block cipher, the key, and the nonce to generate a sequence of m offsets, each offset having n bits;

using the block cipher, the key, the nonce, and the length of the message to generate an n-bit final offset;

for each number i between 1 and m, xoring the i^{th} message block with the i^{th} offset to determine an i^{th} input block;

for each number i between 1 and m, applying the block cipher, keyed by the key, to the i^{th} input block, to determine an i^{th} output block;

for each number i between 1 and m, xoring the i^{th} output block with the i^{th} offset to determine an i^{th} ciphertext block;

concatenating the m ciphertext blocks to determine a ciphertext body;

computing an encoded length by encoding the length of the final fragment as an n-bit string;

xoring the encoded length with the final offset to determine a precursor pad;

computing a pad by applying the block cipher, keyed by the key, to the precursor pad;

xoring the final fragment with a portion of the pad to determine a ciphertext fragment having the same length as the final fragment;

computing a padded ciphertext fragment by appending to the ciphertext fragment a sufficient number of zero bits so that the padded ciphertext fragment has n bits;

computing a checksum by xoring together the m message blocks, the pad, and the padded ciphertext fragment;

computing a precursor full tag by xoring together the checksum and the m^{th} offset;

determining a full tag by applying the block cipher, keyed by the key, to the precursor full tag;

computing a tag as a portion of the full tag; and

defining the ciphertext to be the ciphertext body, the ciphertext fragment, and the tag.

51. (Original) The computer-readable storage medium of claim 50, wherein the i^{th} offset from the sequence of offsets is determined by:

computing a 0^{th} basis offset by applying the block cipher, keyed by the key, to a constant;

for each positive number i, defining the i^{th} basis offset from the prior basis offset by shifting the prior basis offset left one position, and then xoring the resulting value with a constant that depends on the first bit of the prior basis offset;

computing a base offset by applying the block cipher, keyed by the key, to the xor of the 0^{th} basis offset and the nonce;

defining the first offset in the sequence of offsets as the xor of the 0^{th} basis offset and the base offset; and

for each integer i greater than one, defining the i^{th} offset in the sequence of offsets as the xor of the prior offset and the j^{th} basis offset, where j is the number of zero-bits following the last one-bit when the number i is written in binary.

52. (Original) The computer-readable storage medium of claim 50, wherein the final offset is determined by shifting the 0^{th} basis offset one position

to the right, xoring a constant that depends on the last bit of the 0^{th} basis offset, and then xoring the m^{th} offset.

53-60 (Canceled).

61. (Original) An authenticated-encryption apparatus that is configured to use an n-bit block cipher, a key, and an n-bit nonce to encrypt a message into a ciphertext, comprising:

a partitioning mechanism that is configured to partition the message into m message blocks and one final fragment, each message block having n bits and the final fragment having between 0 and n bits;

an offset-generating mechanism that is configured to,

use the block cipher, the key, and the nonce to generate a sequence of m offsets, each offset having n bits, and to

use the block cipher, the key, the nonce, and the length of the message to generate an n-bit final offset;

an xoring mechanism, wherein for each number i between 1 and m, the xoring mechanism is configured to xor the i^{th} message block with the i^{th} offset to determine an i^{th} input block;

an enciphering mechanism, wherein for each number i between 1 and m, the enciphering mechanism is configured to apply the block cipher, keyed by the key, to the i^{th} input block, to determine an i^{th} output block;

wherein for each number i between 1 and m, the xoring mechanism is configured to xor the i^{th} output block with the i^{th} offset to determine an i^{th} ciphertext block;

a concatenating mechanism that is configured to concatenate the m ciphertext blocks to determine a ciphertext body;

a computing mechanism that is configured to compute an encoded length by encoding the length of the final fragment as an n-bit string;

wherein the xoring mechanism is configured to xor the encoded length with the final offset to determine a precursor pad;

wherein the computing mechanism is configured to compute a pad by applying the block cipher, keyed by the key, to the precursor pad;

wherein the xoring mechanism is configured to xor the final fragment with a portion of the pad to determine a ciphertext fragment having the same length as the final fragment;

wherein the computing mechanism is configured to compute a padded ciphertext fragment by appending to the ciphertext fragment a sufficient number of zero bits so that the padded ciphertext fragment has n bits;

wherein the computing mechanism is configured to compute a checksum by xoring together the m message blocks, the pad, and the padded ciphertext fragment;

wherein the computing mechanism is configured to compute a precursor full tag by xoring together the checksum and the m^{th} offset;

wherein the computing mechanism is configured to determine a full tag by applying the block cipher, keyed by the key, to the precursor full tag;

wherein the computing mechanism is configured to compute a tag as a portion of the full tag; and

a defining mechanism that is configured to define the ciphertext to be the ciphertext body, the ciphertext fragment, and the tag.

62-66 (Canceled